Team Description of Team ARAIBO 2007

Yuki Kase, Yuji Hasegawa, Hisanori Mitsumoto, Stefanos Nikolaidis, Tsukasa Kaneko, Keisuke Kobayashi, Takenori Ishibashi, Tamio Arai, Hisashi Osumi, and Kazunori Umeda

1Department of Precision Engineering, School of Engineering, The University of Tokyo
2Department of Precision Mechanics, School of Science and Engineering, Chuo University

aibo@robot.t.u-tokyo.ac.jp
http://araibo.mech.chuo-u.ac.jp

Abstract. Team ARAIBO has taken part in RoboCup since 1999. We have improved the ability of AIBOs with various techniques, which are going to be applied to other robot systems. We introduce them and our latest research briefly in this description.

1 Introduction

Team ARAIBO (Advanced Robotics with Artificial Intelligence for Ball Operation) has taken part in RoboCup since 1999. Our team consists of two universities: The University of Tokyo (since 1999) and Chuo University (since 2002). These universities are located in the center of Tokyo, and are close to each other (about 1.5 km). Our team is composed of three professors, one assistant professor, four graduate students, and three undergraduate students.

We have proposed various novel methods and research developments. Some original attempts in this league have been also introduced by us. For example, we introduced a header action in the 1999 competition at Stockholm [1], while robots of any other team used their legs for carrying the ball at that time. We have brought dynamic programming (DP) [2] in so as to create decision making rules since RoboCup 2001 at Seattle [3–8]. Some studies for enhancing robustness of particle filters have been attempted [9, 10]. Combination of DP and particle filters have been studied as a decision making method under uncertain information [11,12]. Analysis of walking has been also done [13].

These studies do not always bring good result to our team in competition. However, we consider the generality of an algorithm, a method, and a methodology when they are proposed. As a matter of fact, our localization methodology is adopted by two projects that are not directly related to our team [14,15], and we are going to apply it to a sensor room. DP has been also applied to some tasks that are not related to soccer [7,16].

In this description, we introduce our current research in brief. You can find the detail of our soccer program in [17], or in each paper.
2 Studies around DP

One of our interests is to create whole behavior of robots without hand coding. It maybe very important for research for robotics and artificial intelligence. Especially exclusion of heuristics is one of the most interesting topics for us. Our series of studies around dynamic programming (DP) [2] is based on the interesting.

Cooperative Behavior We have attempted to generate cooperative behavior of two robots from a simple code of DP for two years [18]. Though this attempt was unmatured in the last year, now we are on one step short of implementation to actual robots.

![Cooperative Behavior Diagram](image)

**Fig. 1.** Examples of Cooperative Behavior

We show two examples of the robots’ behavior obtained by a 8D state-action map, which has 610 million discrete states. We define Robot1 as the robot that is far from the ball. The other robot is called Robot2. We also define Robot A and B as unique names of the two robots.
When both of the robots cannot observe the ball, RobotA chooses *TurnLeft* and the other chooses *TurnRight* so as to search it.

In Fig. 1(a), RobotA and RobotB started moving from the bottom left corner and in front of the sky-blue goal respectively. At first, RobotA is Robot1 and RobotB is Robot2. The ball is put at the center of the field. In this figure, some important positions of RobotA and RobotB are numbered as Ai and Bi respectively. The numbers are synchronized.

Both of the robots go to the ball until the state (A1, B1). RobotB, however, goes back the way at B1. RobotB hands over the ball to RobotA at the moment though it is Robot2. After that, RobotA becomes Robot2 and RobotB changes into Robot1. As shown this example, Robot2 does not always go to the ball and the role of a robot is never fixed by whether the robot is Robot1 or Robot2. At A2, RobotA kicks the ball. RobotB waits for the kick at B2. After the pass, RobotB reaches B3, which is a final state. At B3, *KickLeft* is chosen as a final shot by an algorithm, which is explained later. The ball is kicked into the goal and the task is finished.

In a precise sense, the pair of RobotA’s kick at A2 and RobotB’s wait at B2 is only a way to reduce the number of steps. However, it is cooperative behavior because the reduction of steps and the role of each robot emerge. The kick by RobotA at A2 can be regarded as *a pass*. In this case, RobotA is the passer and RobotB is the receiver.

Another example with the 8D map is shown in Fig. 1(b). We can see two passes in one trial. RobotB waited for RobotA’s kick at B1 at the first pass. RobotA then waited for RobotB’s kick at A2.

Though the above example is for an old small soccer field, we have obtained a DP result with 3 billion states for the current 6 × 4[m] field. Since the DP result is too huge to install on the robots, compression of this result with a high compression ratio with our VQ compression algorithm [5, 6, 19, 7] will be the next challenge.

**DP with Simulator** As another study of DP, we try creating a state-action map that can consider the shape of ERS-7. Though we cannot release it yet in English, HARIBOTE simulator is used for this attempt.

**Full DP Goalie** We have proposed methods of DP for creating behavior of a goalie at once. In [12], we have shown a goalie that decides its action from its position, its orientation, and the position of the ball. The movie can be seen on our web site. In the site, a movie of a goalie that decides its action not only the five variable, but also the speed of the ball is released. This study is going to be published in Japan. We are now writing an English paper.
3 Time Optimal Control for Quadruped Robots

Though our walking actions are tuned by hand, base motion of walking is derived from result of our study. We show two kinds of the maximum walking speed of our robots in Fig.2 for reference.

Our study walking [13] is based on mechanics. Though learning or genetic algorithms are interesting approach and we want to try, our interest in research is how to calculate time optimal walking of arbitrary quadruped robots.

Our current method is composed of the following processes:

1. A dynamic model of a quadruped walking robot is calculated and a desired path for its body is defined.
2. The trajectory shape of the gravity center of the body is derived from a dynamic constraint of zero moment point (ZMP).

3. Desired torques for leg actuators are obtained by the following processes:
   (a) The time optimal control is designed for supporting legs.
   (b) It is also designed for swinging legs.
   (c) Considering both results, the time optimal walk for the quadruped robot is obtained.

4 HARIBOTE Simulator

We have unveiled a novel simulator, which has been developed in a secret project, in the last open challenge. We could obtain the first place in this challenge, while we were severely panned in the other challenges.

![Fig. 3. Appearance of the Simulator](image)

This simulator, which is named the HARIBOTE simulator, has the potential ability to be the server of RoboCup four-legged simulation league. Of course, such a league is not welcomed by us because we want to play soccer with actual robots. However, it can be an extreme way to prolong the existence of this league until the day new AIBOs arrive.

Related Works Our team has developed a simulator that can simulate characteristics (blur, distortion, noise, ...) of the CMOS camera of AIBOs [20]. UChile’s simulator has re-created the body of ERS-210 in cyber space [21]. Then
simulators for debugging aid have been created by German team [22], ASURA [23], and so on.

![Image](image1)

**Fig. 4.** Interaction of The Robot and The Ball (1,800 meshes in the AIBO)

![Image](image2)

**Fig. 5.** Simulation of CMOS Camera Image

**Specification** Though our new simulator is created from scratch, the above simulators in this league inspire us to develop it. Especially, we have tried combining the nice parts of [20] and [21] in the new simulator.

- real-time computation with a 3 GHz CPU
- superb appearance as shown in Fig.3
- precise calculations of collision and reactive force with polygon meshes (Fig.4)
- inheritance of the ability of CMOS camera simulation from [20] (Fig.5, partially under construction)
**Future Work** Eight AIBOs will be moved simultaneously. In Fig. 6, we show the current version of HARIBOTE simulator. In this figure, six AIBOs are dropped down and their collisions are simulated by physical calculations. The progress of development can be shown in [http://araibo.mech.chuo-u.ac.jp/haribote/](http://araibo.mech.chuo-u.ac.jp/haribote/).

![Fig. 6. Simulation of Multiple Robots](image)

**What is HARIBOTE?** Please make an inquiry to near Japanese.

5 **Self-localization (Expansion Resetting and Combined Resetting)**

We have utilized particle filters for self-localization since 2001. The most different point of our implementation from others may be the way of resetting.

We used the expansion resetting method [10] for the kidnapped robot problems. The sensor resetting method [24] is also used in our code. Figure 7 shows the expansion resetting method and the sensor resetting method [24], which is a popular method in the league.

In the expansion resetting method, the distribution of particles is expanded around the previous distribution. An example is illustrated in Fig.7(c-b). If the expanded distribution still contradicts next observation, it is expanded again. This method can solve the contradiction without drastic change of particle distribution, it can be used for small distance kidnappings.

Since the expansion resetting method is not suitable for long distance kidnappings, we should use the sensor resetting method for such cases. Though the
robot cannot understand the distance of a kidnapping, we have noticed that breadth of the distribution can be used for a threshold of the switching between the expansion resetting method and the sensor resetting method.

Consequently, the methods were chosen with the following if statement:

- if particles exist in a small region: expansion resettings,
- otherwise: sensor resettings.

6 Conclusion

In this report, we have introduced our robotics research along with our effort in RoboCup four legged robot league. There are many things to do for further improvement. In the competition, we think that the following conventional issues are important: faster walking and accurate ball operation. We should reinforce them so as to survive in this league.

References